GENERALIZED PREDICTIVE CONTROL OF DYNAMIC SYSTEMS

A. Hess and Y. C. Jung
Mechanical Engineering
California

Abstract

Generalized Predictive Control (GPC) describes an algorithm for the stable, adaptive control of dynamic systems. In the algorithm, a control input is generated which minimizes a quadratic cost function consisting of a weighted sum of errors between desired and predicted future system outputs and future predicted control increments. The predictions are obtained from an internal model of the plant dynamics. The GPC approach is similar in concept to preview control, which has been discussed in the manual control literature. The GPC algorithm is applied to a simplified rotorcraft terrain following/terrain avoidance problem and its performance is compared to that of a conventional compensatory automatic system in terms of flight path performance, control activity and control law implementation. The potential of the GPC algorithm to serve as a paradigm for the human operator is briefly discussed.

Introduction

Background

In many manual control tasks, the ability of the human operator to "look ahead" or "preview" is a vital strategy in achieving acceptable man/machine performance. Models of human preview control have often employed an "internal model" of the plant dynamics with which the human is presumed to generate predictions of future plant output given current plant state and present and future control inputs e.g., [1,2]. Over the past decade, a technique for the design of automatic controllers, called variously, Model Predictive Heuristic Control, Model Algorithmic Control, or Output Predictive Control, has been introduced which approximates the activity of the human preview controller [3-5].

More recently, Clarke and Zhang [6], and Clarke, et al., [7] have introduced Generalized Predictive Control (GPC) and related it to the earlier approaches of Refs. 3-5 and state-space Linear Quadratic (LQ) designs. It is the GPC approach which is the subject of the research reported herein. Details of the GPC algorithm can be found in Ref. 7, however a brief review of the salient features of the approach will be undertaken in the following sections.

The GPC Algorithm

The plant is modeled in discrete fashion using the so-called Controlled Auto-Regressive Integrated Moving Average model [7]:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \xi(k)/\Delta \qquad (1)$$
$$k = 0, 1, 2, \text{etc.}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator q^{-1}, $y(k)$ and $u(k)$ are output and control variables, respectively, and $\xi(k)$ is an uncorrelated random sequence. Δ represents the differencing operator $1 - q^{-1}$. The actual sampling interval is T, so that, at each sampling instant, the independent variable is kT. Now a prediction of the plant output, given measured output up to time kT and control input $u(k + i)$ for $i \geq -1$, is

$$\hat{y}(k + j|k) = G_j\Delta u(k + j - 1) + F_j y(k) \qquad (2)$$

where

- j = the number of time steps ahead being predicted
- $G_j(q^{-1})$ = E_jB and where E_j results from a recursive solution of the Diophantine relation

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \qquad (3)$$

Here, E_j and F_j are polynomials uniquely defined, given $A(q^{-1})$ and the integer j.

Now a predictive control law can be defined as that which minimizes the cost function given by

$$J(N_1,N_2) = E\left[\sum_{j=N_1}^{N_2}[\hat{y}(k + j) - w(k + j)]^2 + \sum_{j=1}^{N_2}\lambda(j)[\Delta u(k + j - 1]^2\right] \qquad (4)$$

where

- N_1 = the minimum costing horizon
- N_2 = the maximum costing horizon
- $w(k)$ = the desired value of the output y at the k^{th} sampling instant
- $\lambda(j)$ = a control weighting sequence

Equation (4) is concerned only with a subset of future time defined N_2T secs into the future and is dependent upon data up to time kT. Note how the control is implemented: The optimal control at the first sampling instant is applied and the minimization of J is repeated at the next sample. Also note that the cost on the control is over all future control inputs which effect the outputs included in J. This control law can be classified as Open-Loop-Feedback-Optimal with an autoregressive disturbance process [7]. The authors feel that this control philosophy is similar to that which the human operator employs when controlling plants for which desired future output can be defined. Examples are automobile driving or aircraft flight path control in near-earth flight.

Significant reductions in the order of the matrices involved in computing the optimal control can be made by requiring that, after an interval $NU < N_2$, projected

control increments are assumed to be zero, i.e.,

$$\Delta u(k + j - 1) = 0 \qquad j > NU \tag{5}$$

where NU is called the "control horizon". This is equivalent to placing infinite weights on control changes after some future time. In addition to computational simplifications, introduction of the control horizon also allows the stable control of non-minimum phase plants [7].

With the introduction of the control horizon, the prediction equations become

$$\hat{\underline{y}} = \underline{G}_1 \tilde{\underline{u}} + \underline{f} \tag{6}$$

where

$$\hat{\underline{y}} = [\hat{y}(k + 1), \hat{y}(k + 2), \ldots . \hat{y}(k + N)]^T$$

$$\tilde{\underline{u}} = [\Delta u(k), \Delta u(k + 1), \ldots . \Delta u(k + N - 1)]^T$$

$$\underline{f} = [f(k + 1), f(k + 2), \ldots . f(k + N)]^T$$

N = output horizon = N_2 here.

$$\underline{G}_1 = \begin{bmatrix} g_0 & 0 & \ldots & 0 \\ g_1 & g_0 & & \\ \cdot & \cdot\cdot & \ldots & 0 \\ \vdots & \cdot & & g_0 \\ \vdots & \cdot & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix} \tag{7}$$

with $f(k + j)$ being that component of $y(k + j)$ composed of signals which are known at time kT [7], and the g_i are elements of the polynomial $G_j(q^{-1})$, itself obtained from the recursive Diophantine relation (3). The corresponding control law is given by

$$\tilde{\underline{u}} = (\underline{G}_1^T \underline{G}_1 + \lambda \underline{I})^{-1} \underline{G}_1^T (\underline{w} - \underline{f}) \tag{8}$$

where

$$\underline{w} = [w(k + 1), w(k + 2), \ldots . w(k + N)]^T$$

The matrix involved in the inversion above is of dimension NU x NU. Equation (8) and the pertinent relations preceding it define the GPC algorithm. Although not considered here, the GPC algorithm can be made adaptive by the inclusion of a "standard" recursive least-square parameter estimator [8]. Some theoretical stability results are presented in Ref. 7 by relating GPC to state-space LQ control laws. The reader is referred to this reference for details.

A number of parameters are obviously available as design variables in applications of the GPC algorithm. They are: The minimum and maximum costing horizons, N_1 and N_2, the control horizon NU, and the control weighting sequence $\lambda(k)$. The role played by these parameters is best demonstrated by means of the flight control example of the following section.

The on-line computational requirements of the GPC algorithm for cases in which no adaptation is occurring are very minimal since all major computations including the matrix inversion of Eq. (8) can be performed off-line. Thus, on-line computations are limited to the matrix multiplications shown in Eq. (8), with $N = N_2$.

Applications to Flight Control

Introduction

Terrain-following/terrain-avoidance (TF/TA) flight offers a significant challenge to the designers of automatic flight control systems. The response requirements of these systems imply relatively high bandwidth outer loop command following characteristics which are difficult to obtain using classical design techniques. The ability of the human pilot to succesfully complete such tasks has led to the investigation of pertinent preview control models for near-earth flight [9]. The similarity between the philosophy of these models and that of the GPC approach led to a consideration of the latter algorithm as a candidate for automatic flight path control in the TF/TA task. Indeed, Reid, et al., [5], have applied an Output Predictive algorithm to a terrain following flight control task. Conceptually at least, this algorithm is a special case of GPC as it considers the control input to be held constant over some number of sampling intervals, then provides a least-square control solution which minimizes a cost function similar to Eq. (4), but with no weighting on control inputs, a minimum output horizon of zero, and a control horizon matching the maximum output horizon. The necessity of holding the control input constant over a number of sampling intervals arose in ensuring output stability.

In conducting some preliminary evaluations of the Output Predictive algorithm for the height control task to be considered here, performance was, in general, unsatisfactory. The necessity of holding the control input constant for multiples of the sampling interval coupled with the lack of control weighting in the cost function led to unrealistic control inputs, i.e., control signals which resembled relay-like functions alternating between large positive and negative amplitudes in all applications. For this reason, the Output Predictive algorithm was eschewed in favor of the GPC system to be described.

Simplified Rotorcraft Vertical Dynamics

Figures 1-3 show the three "plants" which were utilized in this study. They all involve a simplified rotorcraft "bare-airframe" vertical velocity to collective input transfer function given by

$$\frac{\dot{h}}{\delta}(s) = \frac{-(s - 20)}{(s + 1)(s + 20)} \doteq \frac{e^{-0.1s}}{(s + 1)} \tag{9}$$

The introduction of the first-order Pade' approximation to the time delay offers an interesting challenge to the control algorithm since it involves non-minimum phase dynamics. Figure 1 represents a "bare-airframe" in which the control input for the GPC algorithm will be collective control. Figure 2 represents the bare-airframe with a vertical velocity control loop closed about it. Here, the control input for the GPC algorithm will be commanded vertical velocity $\dot{h}_c$. The effective plant for this case will be

$$\frac{\dot{h}}{\dot{h}_c}(s) = \frac{-4(s + 0.5)(s - 20)}{s(s^3 + 17s^2 + 98s + 40)} \tag{10}$$

$$= \frac{-4(s + 0.5)(s - 20)}{s(s + 0.44)[s^2 + 2(0.87)(9.52)s + 9.52^2]}$$

Finally, Fig. 3 represents the bare-airframe with velocity and height control loops closed. Here the control input for the GPC algorithm will be commanded height. The effective plant for this case will be

$$\frac{h}{h_c}(s) = \frac{-4(s + 0.5)(s - 20)}{s^4 + 17s^3 + 94s^2 + 118s + 40} \quad (11)$$

$$= \frac{-4(s + 0.5)(s - 20)}{(s + 1)(s + 0.579)[s^2 + 2(0.93)(8.31)s + 8.31^2]}$$

The rationale for selecting the dynamic systems of Figs. 1-3 was that they represented the range of possible levels of GPC utilization in a typical flight control application from inner-loop control actuator commands in Fig. 1 to outer-loop height guidance commands in Fig. 3.

Terrain Following/Terrain Avoidance

The commanded vertical flight path for this application was actually a time history similar to that utilized in Ref. 5, represented as a sum of sinusoids

$$h_c = 20[\sin(.05(2\pi t)) + \sin(.06(2\pi t)) + \sin(.08(2\pi t))] \text{ ft} \quad (12)$$

Equation (12) can be thought of a representing a commanded flight path which would be provided by an on-board computer in a TF/TA task. In implementing the GPC algorithm, the "desired" output or vehicle path was an exponential curve which continuously defined a smooth "capture" trajectory from the vehicle's present position to the command of Eq. (12). This capture trajectory was given by

$$h_c'(j + k) = h_c(k + j) - \exp(-\tau_e j)[h_c(k + j) - h(k)] \quad (13)$$

$$j = 1, 2, \ldots, N_2$$

Although the time constant τ_e could serve as another design variable in the GPC algorithm, it was maintained at 0.5 secs for this study. Thus, the time to 50% and 95% amplitudes for the trajectory of Eq. (13) was 0.14 and 0.6 secs, respectively. These values were deemed acceptable for this vehicle and task.

Figure 4 shows the performance of the system of Fig. 3 without GPC and with the command trajectory of Eq. (12) serving as the system input. This serves as a benchmark system for GPC performance comparisons as it prepresents the performance of a "classical" multi-loop control design with fairly high loop bandwidths. Note the height errors exceed 20 ft in some instances. This classical design has been discretized with a 0.1 sec sampling interval so that it is comparable to the GPC implementation.

Figures 5-7 show the performance of the GPC systems. The command and actual vehicle trajectories (dashed and solid lines, respectively) are indistinguishable in these figures because of the excellent tracking performance. This performance is indicated by the small height errors, where, with the exception of the initial and final transients, they are less tha 1 ft in magnitude. The transients are due to the abrupt initiation and termination of the height command at the zero crossings of the sum of sinusoids at the begininning and end of the simulation. The figure parts labeled "GPC Input" represent the "control" as provided by the GPC algorithm (u in Eq. (1)), and this input varies from the systems of Figs. 1 through 3. The excellent performance of the GPC algorithm is evident in all the systems with performance deteriorating slightly as one moves from the system of Fig. 1 to that of Fig. 3. The GPC parameters for all the applications were

N_1 = minimum output horizon = 1 (0.1 secs)
N_2 = maximum output horizon = 50 (5 secs)
NU = control horizon = 20 (2 secs)
λ = control weighting sequence = 0.2

These values were obtained by a trial and error procedure.

It was of interest to investigate the robustness of the GPC algorithm as regards the quality of the "internal model" which was used in the Diophantine relation (3). To this end, a brief investigation was conducted on the system of Fig. 1 in which the dynamics of Eqs. (9) were not changed in the digital simulations, but the "internal model" of these dynamics in the GPC algorithm were given by

$$\frac{\dot{h}}{\dot{q}}(s) = \frac{1}{(s + 1)} \quad (14)$$

i.e., the Pade' approximated time delay was omitted. For all practical purposes, the results were identical to those of Fig. 5 with no change in the GPC design parameters necessary. This is an encouraging result, as it indicates that inaccurate modeling of system delays or higher frequency system dynamics, will not have a detrimental effect upon GPC performance.

Manual Control Applications

Although not pursued in this study, the application of the GPC algorithm to the description of manual control tasks in which desired future output can be defined apprears promising. Tasks which immediately come to mind are automobile driving and aircraft near-earth flight. The inclusion of weightings on control rate in the cost function of Eq. (4) as is typically done in the Optimal Control Model of the human operator [10], can be accomplished by suitable modification of the GPC algorithm [7]. The basic format of the GPC approach, with its output and control horizons, its internal model, and its output (as opposed to state) feedback structure make it a worthy candidate for future research in the manual control area.

Conclusions

Based upon the analyses performed to date, the following conclusions can be drawn:

1.) The GPC algorithm offers tracking performance superior to classical multi-loop control system designs. In the simplified TF/TA task studied here, an order of magnitude reduction in absolute height errors was achieved.

2.) The GPC algorithm can be introduced with equal ease and success at a number of different points in a control hierarchy. In the examples studied here, GPC produced optimal control policies where "control" was defined from inner-loop actuator commands to outer-loop guidance commands.

3.) The on-line computational requirements for the non adaptive GPC applications are minimal.

4.) A limited examination of the effects of inaccuracies in the GPC internal model upon GPC performance indicates that errors in the estimation of plant time delay or higher frequency dynamics have minimal effect upon performance.

Acknowledgment

This work was supported by a grant from the Aircraft Guidance and Navigation Branch of NASA Ames Research Center.

References

[1] Sheridan, T. B., "Three Models of Preview Control," IEEE Transactions on Human Factors in Electronics, Vol. HFE-7, No. 2, June 1966, pp. 91-102.

[2] Kelley, C. E., Manual and Automatic Control, Chap. 15, Wiley, New York, 1968.

[3] Richalet, J., Rault, A., Testud, J. L., Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," Automatica, Vol. 14, No. 5, Sept 1978, pp. 413-428.

[4] Rouhani, R., and Mehra, R. K., "Model Algorithmic Control (MAC): Basic Theoretical Properties," Automatica, Vol. 18, No. 4, July 1982, pp. 401-414.

[5] Reid, J. G., Chaffin, D. E., and Silverthorn, J. T., "Output Predictive Algorithmic Control: Precision Tracking with Applications to Terrain Following," Journal of Guidance and Control, Vol. 4, No. 5, Sept.-Oct. 1981, pp. 502-509.

[6] Clarke, D. W., and Zhang, L., "Long-Range Predictive Control Using Weighting-Sequence Models," IEE Proceedings, Vol. 134, Pt. D., No. 3, May 1987, pp. 187-195.

[7] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control, Parts I and II," Automatica, Vol. 23, No. 2, March 1987, pp. 137-160.

[8] Gelb, A., Applied Optimal Estimation, MIT Press, Cambridge, 1974.

[9] Hess, R. A., and Chan, K. K., "A Preview Control Pilot Model for Near-Earth Maneuvering Helicopter Flight," to appear in Journal of Guidance, Control and Dynamics, Vol. 11, No. 1, Jan.-Feb. 1988.

[10] Hess, R. A., "Feedback Control Models," in Handbook of Human Factors, Wiley, New York, G. Salvendy, Ed., 1987, Chap. 9.5.

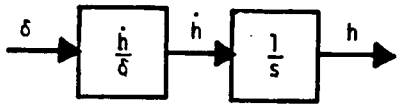
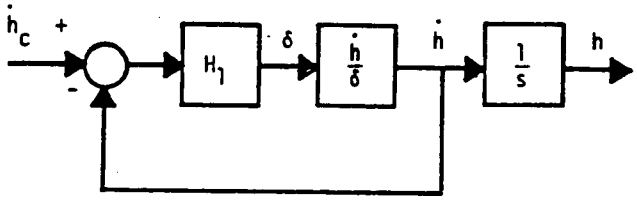
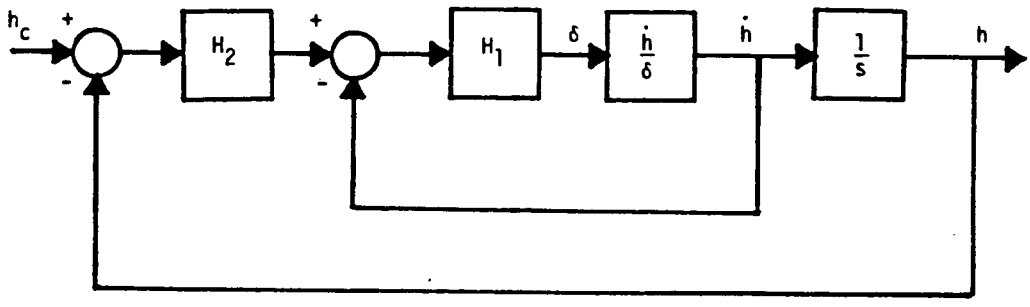


Fig. 1 Bare-airframe



$$H_1 = \frac{4(s + 0.5)}{s}$$

Fig. 2 Bare-airframe with vertical velocity control loop



$$H_1 = \frac{4(s + 0.5)}{s} \qquad H_2 = 1.0$$

Fig. 3 Bare-airframe with vertical velocity and height control loops

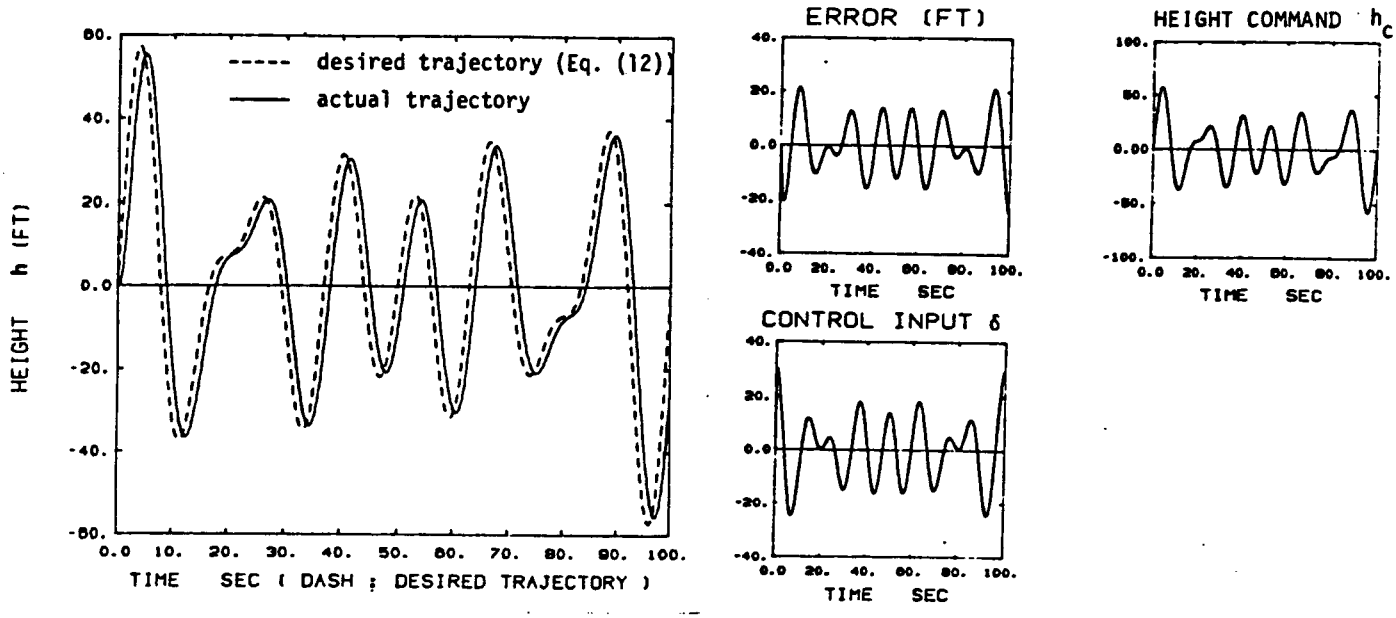


Fig. 4 Terrain following performance of system of Fig. 3

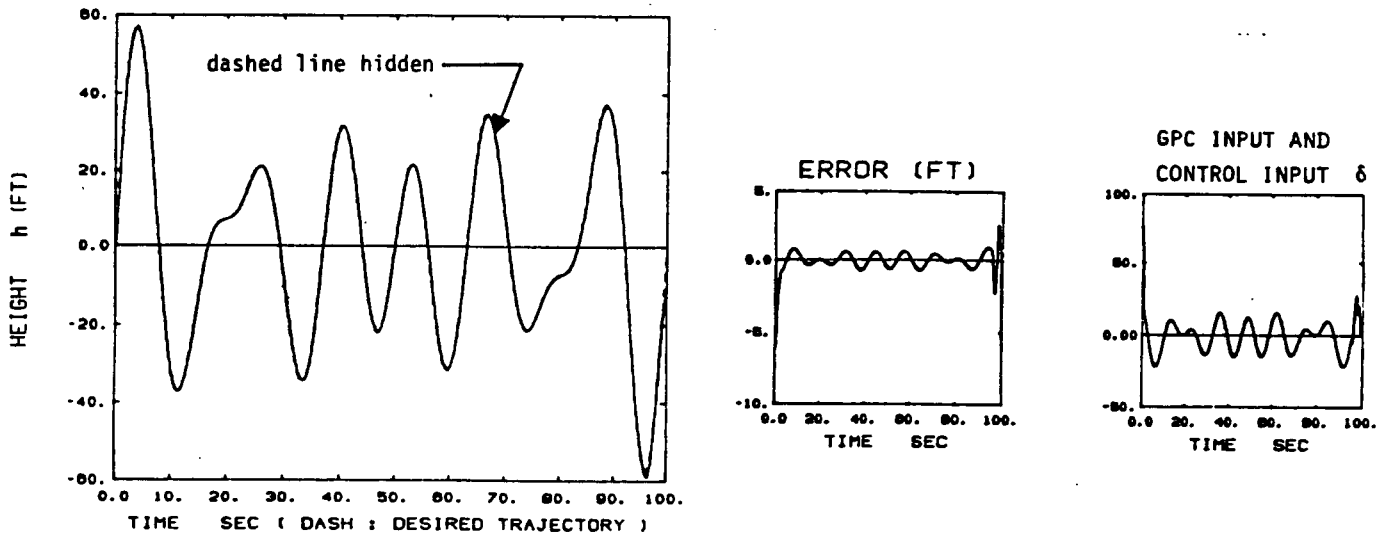


Fig. 5 Terrain following performance of system of Fig. 1 with GPC

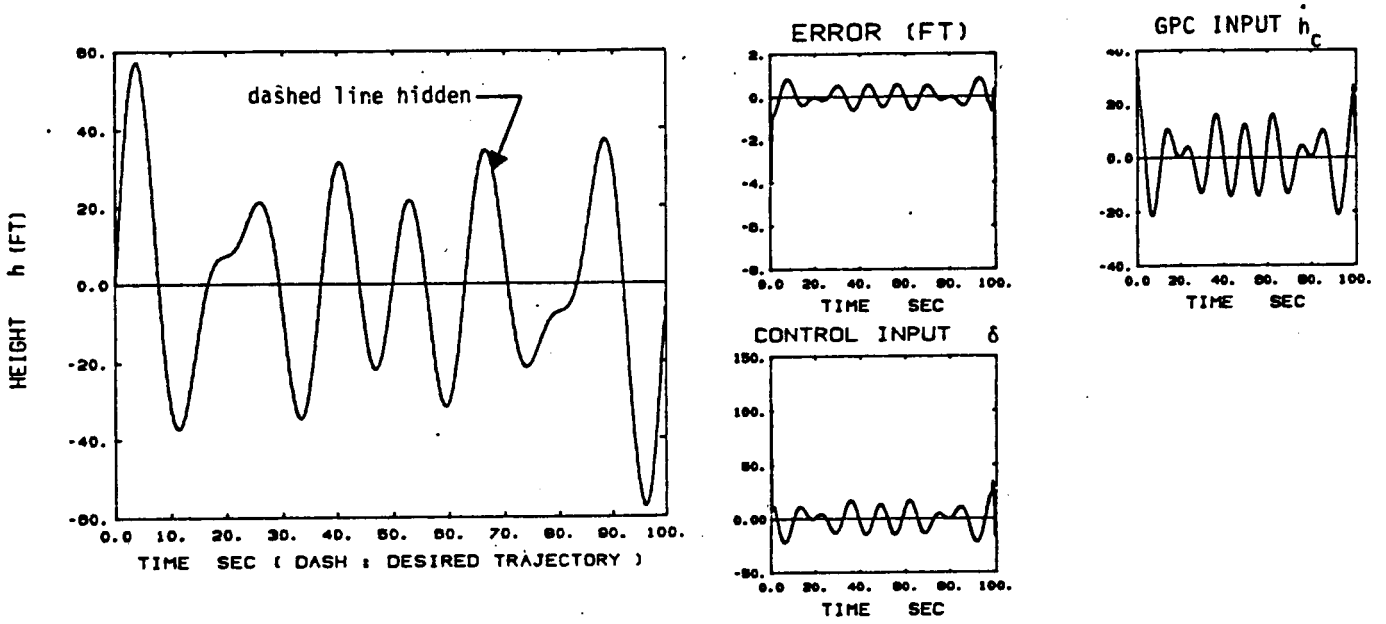


Fig. 6 Terrain following performance of system of Fig. 2 with GPC

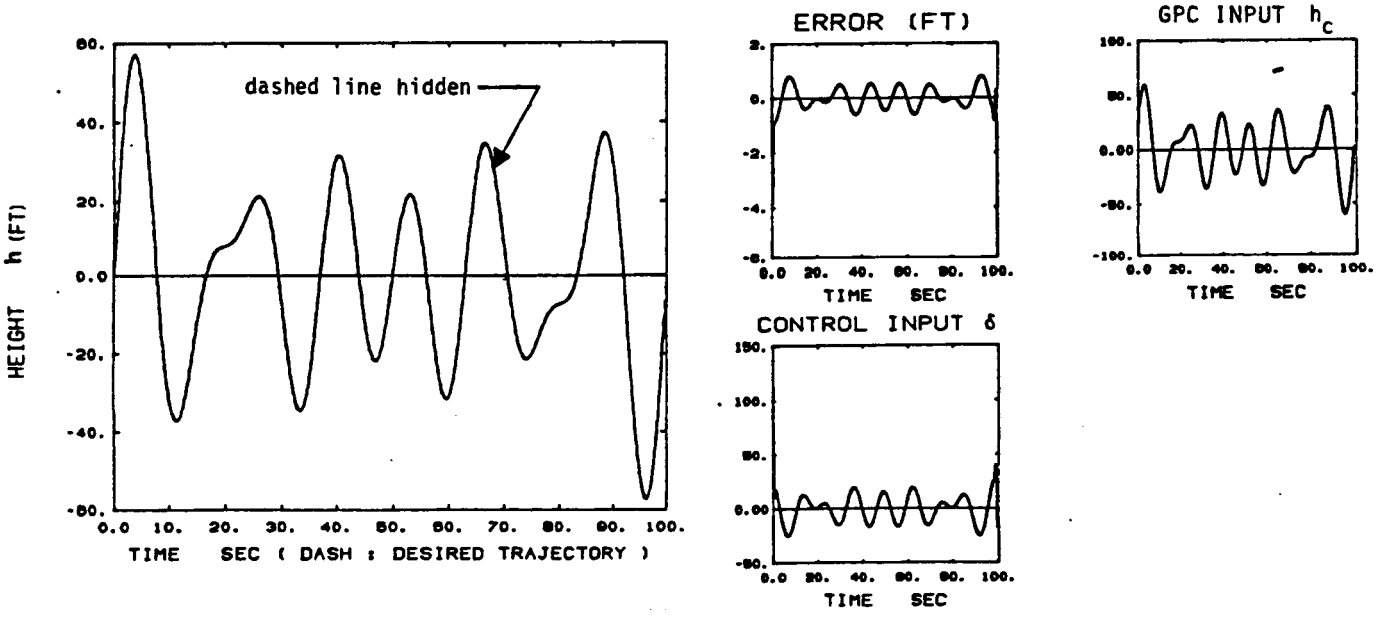


Fig. 7 Terrain following performance of system of Fig. 3 with GPC